



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/750,227	12/31/2003	Stanislav Sosnovsky	EMC03-22(03111)	6637
58404	7590	01/06/2009	EXAMINER	
BARRY W. CHAPIN CHAPIN INTELLECTUAL PROPERTY LAW, LLC WESTBOROUGH OFFICE PARK 1700 WEST PARK DRIVE, SUITE 280 WESTBOROUGH, MA 01581			PRICE, NATHAN E	
		ART UNIT		PAPER NUMBER
		2194		
		MAIL DATE	DELIVERY MODE	
		01/06/2009	PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)
	10/750,227	SOSNOVSKY ET AL.
	Examiner	Art Unit
	NATHAN PRICE	2194

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 16 October 2008.
- 2a) This action is **FINAL**. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-9, 12-17, 20-28, 30, 31, 33-36, 38-49 and 51-54 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-9, 12-17, 20-28, 30, 31, 33-36, 38-49 and 51-54 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ . |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ . | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| | 6) <input type="checkbox"/> Other: _____ . |

DETAILED ACTION

1. Claims 1 – 9, 12 – 17, 20 – 28, 30, 31, 33 – 36, 38 – 49 and 51 – 54 are pending.
2. This Office Action is in response to communications received 16 October 2008. Previous objections and rejections not included in this Office Action have been withdrawn.

Response to Arguments

3. Applicant's arguments, see REMARKS, filed 16 October 2008, with respect to rejections under 35 USC 112, 1st ¶, have been fully considered and are persuasive. The rejections under 35 USC 112, 1st ¶, of claims 1 – 9, 12 – 17, 20 – 28, 30, 31 and 33 – 49 have been withdrawn.
4. Applicant's arguments filed 16 October 2008 regarding rejections under 35 USC 103 have been fully considered but they are not persuasive.
5. Regarding claim 1, Applicant argues Silberschatz teaches actions initiated by an OS and not an application. It is not clear what limitations of claim 1 are being argued with respect to asserting Silberschatz teaches actions initiated by an OS and not an application. However, Silberschatz teaches applications can subscribe to events (p. 641 ¶5).

6. Applicant references CORBA in the arguments to explain why the references fail to render the claims, specifically claim 1, unpatentable. In response to applicant's argument that the references fail to show certain features of applicant's invention, it is noted that the features upon which applicant relies (i.e., CORBA) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).

7. Applicant's other arguments appear to be focused on arguing that Frank teaches activation of threads. For example, Applicant argues Frank fails to teach component invocation. However, Frank is not relied upon to teach component invocation. Silberschatz is relied upon to teach component invocation (§9.2 ¶1 – 4). Frank is cited to teach features of event delivery, not component invocation. Not teaching component invocation does not prevent one of ordinary skill in the art to use the event processing features of Frank to modify event processing taught by Silberschatz.

8. See current rejections regarding new limitations and claims added by amendment.

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 1 – 9, 12 – 17, 20 – 28, 30, 31, 33 – 36, 38 – 49 and 51 – 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Silberschatz (see PTO-892 mailed 02 May 2007) further in view of Frank et al. (US 2004/0250254 A1; hereinafter Frank).

10. As to claim 1, Silberschatz teaches a method for processing events in a managed information system comprising:

receiving an event subscription containing an event identity for an event, the event corresponding to reportable occurrences in the managed information system (page 641 ¶ 5 – 6);

associating the event identity with an event handler responsive to the event by creating a mapping of the event identity to the event handler (page 641 ¶ 5 – 6);

receiving a publication of the event (page 641 ¶ 5 – 6); and

traversing, in response to the publication, the mapping of the event identity to an indication of the corresponding associated handler, the traversing operable to enable the module including the event handler if the module is disabled at the time of publishing the event (page 641 ¶ 5 – 6; page 407 ¶ 3), such that

the enabling of modules corresponds to activation of a corresponding component by an activation mechanism; and disabling corresponding to deactivation of the corresponding component by the activation mechanism (section 6.1.4; 9.2 ¶ 1 – 4),

the activation and deactivation operations operable to reduce memory consumption by inactive components and provide selective invocation to maintain availability of the component, the enabling and disabling performed at a level of granularity of the modules, each of the modules corresponding to a component and operable be enabled and disabled by activation and deactivation of the corresponding component (section 6.1.4; 9.2 ¶ 1 – 4).

11. Silberschatz fails to teach defining a plurality of events, a genericizing reference, an event specific class and event variables as claimed. However, Frank teaches defining a plurality of events, the plurality of events associated with a genericizing reference, the genericizing reference inclusive of the plurality of events and each of the events associated with an event specific class having event data indicative of event specific parameters (¶ 90 – 92), each of the events defined independently of an underlying delivery infrastructure (¶ 104, 146); and

the event data includes event variables generated and passed by the publisher of the event and subscriber instantiated variables generated by the state information of the subscriber (¶ 90 – 92, 103).

12. Silberschatz and Frank also teach selectively enabling, if the module including the corresponding event handler is disabled, the module for enabling the event handler for receipt and subsequent processing of the published event (Silberschatz: page 641 ¶ 5 – 6; section 6.1.4; 9.2 ¶ 1 – 4) (Frank: ¶ 15, 18, 24, 75, 76, 82).

13. It would have been obvious to one of ordinary skill in the art at the time Applicant's invention was made to combine these teachings because both teach events and how to deliver the events.

14. As to claim 2, Silberschatz fails to teach identifying as claimed. However, Frank teaches identifying, using the associated event identity, the particular handler corresponding to the subscribed event in the enabled module including the handler; invoking, using the state of the enabled module, the event handler (¶ 18).

15. As to claim 3, Silberschatz and Frank teach traversing further comprises: receiving the event publication according to a genericizing reference; and identifying an event specific class corresponding to the event, the event specific class transparent to the mapping via the genericizing reference and operative to distinguish the received events from other events (Frank: ¶ 96, 176).

16. As to claim 4, Silberschatz and Frank teach traversing further comprises determining if a particular module including the corresponding event handler is enabled (Frank: ¶ 15, 18, 75).

17. As to claim 5, Silberschatz and Frank teach, following selective enabling of the module containing the corresponding handler: determining the mapping of the enabled module and corresponding event handler; and invoking the module including the corresponding event handler via the mapping (Frank: ¶ 90, 94).

18. As to claim 6, Silberschatz and Frank teach traversing further comprises identifying the event in a persistent event mapping, the persistent event mapping indicative of modules containing event handlers associated with the event; and dispatching, in the identified modules, the associated event handlers (Frank: ¶ 75, 81 – 84, 90, 99, 255).

19. As to claim 7, Silberschatz and Frank teach dispatching further comprises: selectively receiving an enablement indication in response to traversing in the persistent event mapping; identifying, in a local event mapping for the enabled module, subscriber entities including handlers associated with the mapped event; and invoking, in the mapped module, the identified subscribers including associated handlers (Frank: ¶ 18, 75, 94 – 99).

20. As to claim 8, Silberschatz teaches the persistent mapping of the event is operable to maintain the event independently of individual modules referencing the event, the independent maintenance operable to avoid copy constructors of the event for enabling successive references to the same event (page 23 ¶ 3).

21. As to claim 9, Silberschatz and Frank teach disabling a publishing component performing the publishing prior to invoking the module including the event handler; and completing the invocation of the corresponding handler while the publishing component remains disabled (Silberschatz: page 407 ¶ 2) (Frank: ¶ 20, 101 – 107).

22. As to claim 12, Silberschatz teaches enumerating a plurality of events, wherein the event further comprises the plurality of events associated with the common genericizing reference inclusive of the plurality of events, and wherein receiving the event subscription avoids event specific code generation of code and code fragments associated with the specific event (page 23 ¶ 3).

23. As to claim 13, Silberschatz teaches the common genericizing reference and associated event specific class avoids event specific stubs and references related to the event specific class (page 23 ¶ 3).

24. As to claim 14, Silberschatz teaches a subscribing software entity issuing the received event subscription becomes disabled following the subscription until an occurrence and subsequent publication of the event (page 641 ¶ 5 – 6; page 417 ¶ 3).

25. As to claim 15, Silberschatz teaches publication of the event is operable to enable a plurality of subscribing software entities, each subscribing entity including a particular responsive event handler for handling that event (page 641 ¶ 5 – 6).

26. As to claim 16, Frank teaches traversing further comprises indexing, in the persistent mapping via the event identity, a persistent reference to the modules including the event handlers associated with the event, the persistent reference operable to identify a handler independently of enablement of the module containing the associated event handler (¶ 75, 99).

27. As to claim 17, Silberschatz and Frank teach associating the event identity by creating a mapping with the event handler further comprises: creating, via a component event service, a local mapping entry in the component event map having a reference to the subscriber entity including the corresponding event handler, and creating a persistent mapping entry corresponding to the component including the corresponding event handler, the persistent mapping entry operable to trigger selective enablement of the handling component by a plurality of subscribing entities, wherein mapping further comprises: identifying at least one of the persistent mapping entries corresponding to

the published event, each of the mapping entries indicative of a module; and identifying, via the local event map in the indicated modules, a plurality of subscribers including the corresponding event handlers in the identified modules associated with the event (Silberschatz: page 23 ¶ 3) (Frank: ¶ 18, 75, 99).

28. As to claim 20, Silberschatz and Frank teach activation and deactivation further comprises identifying, in a component server in communication with the shared memory portion, when to activate and deactivate components based on information in the persistent event map in the shared memory portion, and further for determining when to store the information in the component server rather than shared memory if no other component servers reference the information (Silberschatz: page 23 ¶ 3) (Frank: ¶ 18, 75, 99).

29. As to claim 21, Silberschatz teaches each of the modules is operable to include a plurality of threads, and disabling is performed by a thread manager operable to gracefully terminate each of the threads prior to deactivation, deactivation occurring by informing each of the threads of the termination and computing when each thread has attained a termination point (section 4.1.4; 6.1.4; 9.2 ¶ 1 – 4).

30. As to claim 22, Silberschatz teaches associating the event identity with an event handler occurs in a native language of the event handler and corresponding subscriber,

and avoids a corresponding definition in an external interface language, the external interface language for generating event specific code (section 12.3.3; 3.3 ¶ 1 – 3).

31. As to claim 23, Silberschatz teaches the external interface language is an Interface Definition Language (section 15.4; page 519).

32. As to claim 24, Silberschatz and Frank teach associating the event identity with an event handler further comprises generating a local mapping via a component event service identifying a subscribing entity including an event handler corresponding to the event identity, and generating a persistent event mapping identifying the module including the event handler corresponding to the event identity (Silberschatz: page 23 ¶ 3) (Frank: ¶ 18, 75, 99).

33. As to claim 25, Silberschatz and Frank teach selectively generating the persistent event mapping via a strategized allocator if the associating of the event identity occurs in a single module (Silberschatz: page 23 ¶ 3) (Frank: ¶ 18, 75, 99).

34. As to claims 26 and 27, see the rejection of claims 1, 2 and 17. Regarding the smart pointer and reference counting semaphore see Silberschatz section 22.4.1.

35. As to claim 28, Frank teaches the subscribing entity, publishing entity, and handling entity are user software entities responsive to the local event service for execution, activation, and inactivation (¶ 18, 75).

36. As to claims 30 – 31, see the rejection of claims 1 and 17.

37. As to claim 33 – 34 and 51, see the rejection of claims 1, 2, 6, 7, 14, 17 and 26.

38. As to claims 35, 36 and 38 – 48, see the rejection of claims 1 – 8, 16, 17, 21, 24 and 25.

39. As to claim 49, see the rejection of claim 1.

40. As to claim 52, Silberschatz teaches activating further comprises:
loading a module, the module containing the instructions defining the component, from a nonexecutable memory area to an executable memory area for execution of the instructions (§9.2 ¶ 1 – 4); and
deactivating further comprises unlinking the loaded module by disassociating the loaded module from the executable memory area (§9.2 ¶ 1 – 4).

41. As to claim 53, Silberschatz teaches the loaded modules are sequences of instructions defining at least one thread, all of the threads collectively activated and

deactivated via the component defined by the module (§4.1.4; §9.2 ¶ 1 – 4).

42. As to claim 54, Silberschatz teaches selective activation further comprises:
- detecting, from a loaded component defined by a detecting software entity, occurrence of an instance corresponding to an event handler (page 641 ¶ 5 – 6; §9.2 ¶ 1 – 4); and
- invoking the event handler by activating the component including the event handler (page 641 ¶ 5 – 6; §9.2 ¶ 1 – 4).

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to NATHAN PRICE whose telephone number is (571)272-4196. The examiner can normally be reached on 8:00am - 4:30pm, Monday - Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Li B. Zhen/
Primary Examiner, Art Unit 2194

NP